classification

Machine learning → Machine learning is the process, computers are given the ability to learn to make decision from data.

without being explicitly programmed!

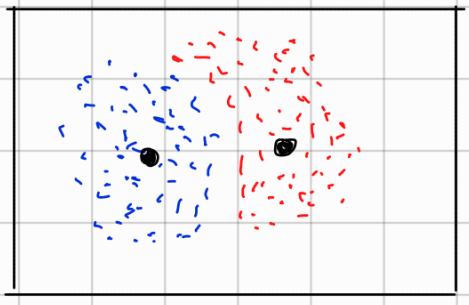
email → spam / not spam

unsupervised learning

→ uncovering hidden pattern from unlabeled data.

→ Example,

Grouping customers into distinct categories (clustering)



Cluster Analysis of customer churn.

Supervised learning

→ The predicted values are known

→ predict the target values of unseen data, given the features.

classification
(target variable consists of categories)

Regression
(target variable is continuous)

ex:
fraudulent transaction & non-fraudulent transaction

ex:
no. of. rooms, size of property to predict the target variable (price)

feature → Independent variable.

target variable → dependent variable.

Requirements of Supervised learning

* No missing data
* Data in numeric format
* Data stored in pandas Dataframe or Numpy array.

→ perform Exploratory Data Analysis (EDA) first.

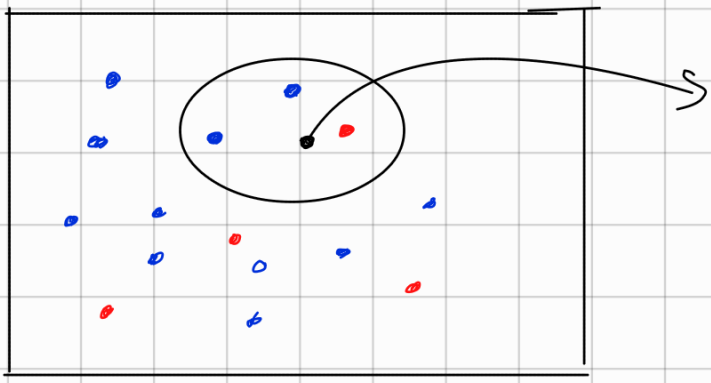
classifying labels of unseen data

1. Build a model.
2. Model learns from the labeled data we pass to it
3. pass unlabeled data to the model as input.
4. Model predict the labels of the unseen data.

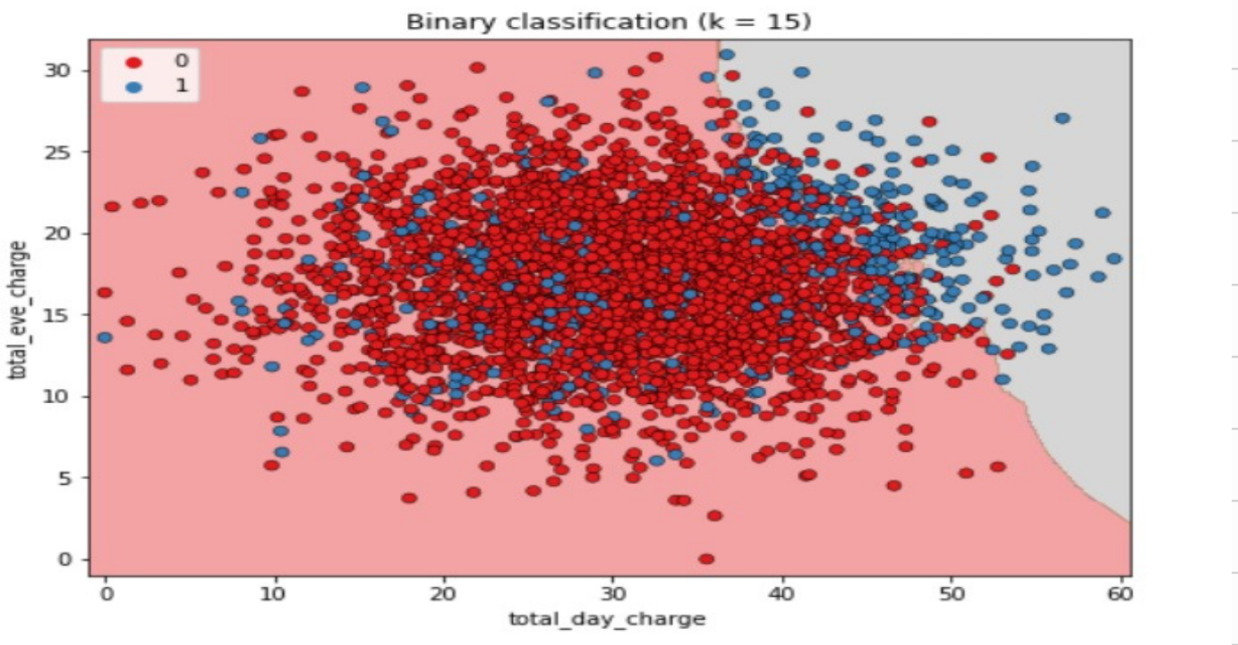
(labeled data = training data)

K- Nearest Neighbors

→ predict the label of a data point by
→ looking at the 'k' closest labelled data points.
→ taking majority voting.



→ KNN creates a decision Boundary to predict the value



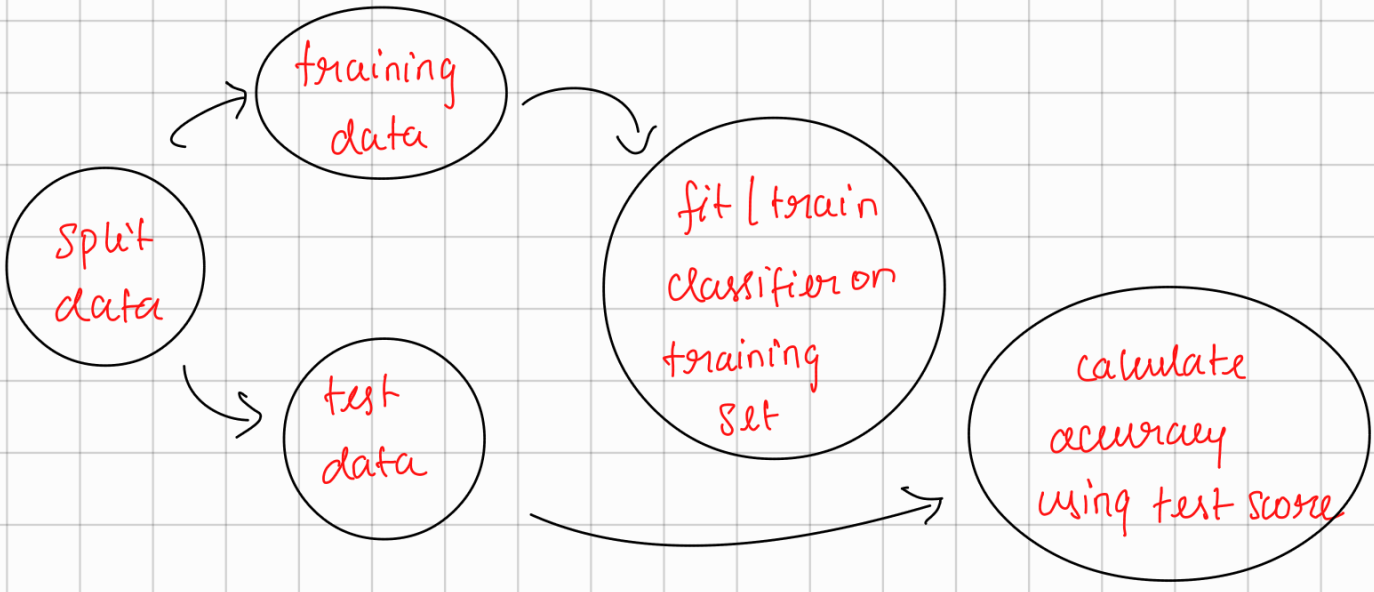
measuring Model performance

→ In classification, accuracy is a commonly used metric.

$$\text{Accuracy} = \frac{\text{correct prediction}}{\text{total observation}}$$

→ could compute accuracy on the data used to fit the classifier

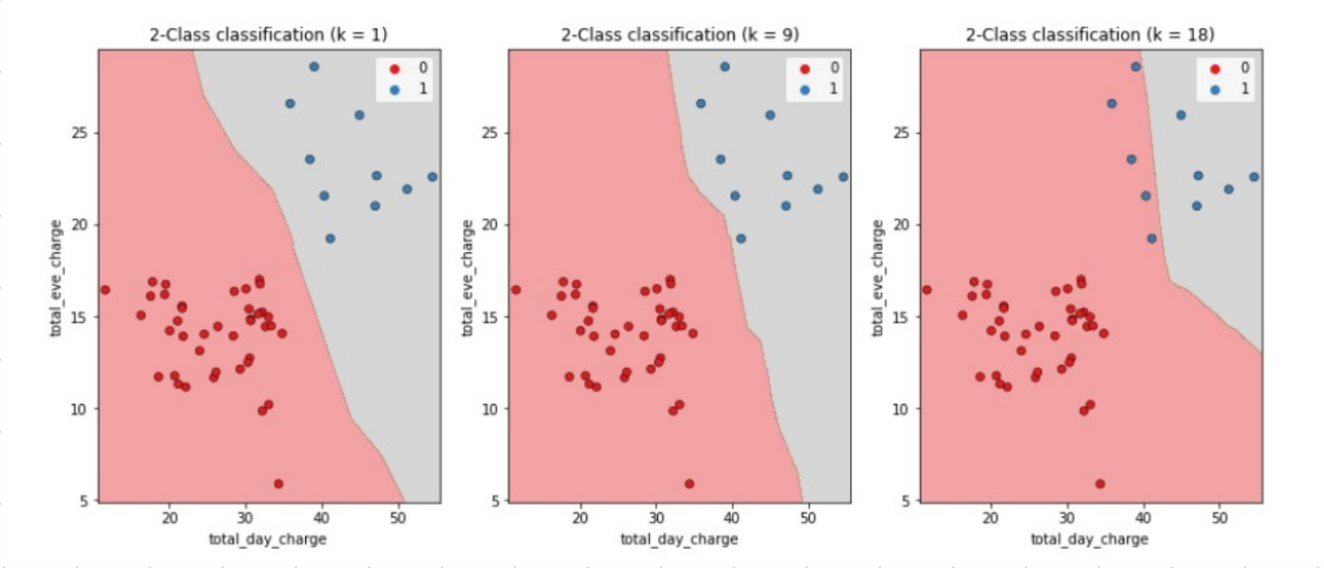
→ not indicative of ability to generalize.



Model complexity

→ larger (k), less complex Model, can cause underfitting

→ Smaller k, more complex model, can lead to overfitting.



Introduction to Regression

→ target variable typically has continuous values.

→ predicting blood glucose level.

→ creating feature and target arrays.

→ Making predictions from a single feature

→ Plotting glucose vs Body mass index.

→ fitting a regression (linear) model.

Basics of linear regression

Regression Mechanics

$$y = ax + b$$

→ simple linear regression uses one feature.

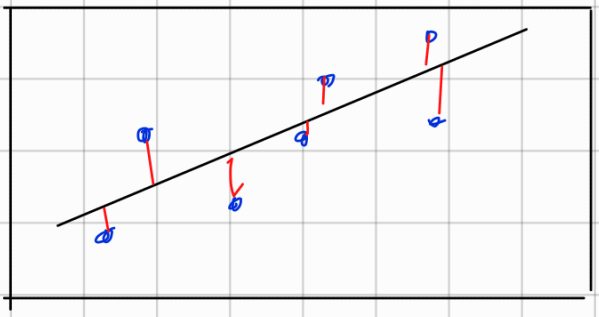
a, b → parameters / coefficients of the model — Slope & Intercept.

x → single feature.

define the error function for any given line and choose the line that minimize the error function.

Error / loss / cost function.

Vertical distance: (residual)



loss function.

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y})^2$$

(residual sum of squares)

Ordinary least squares (OLS): Minimize RSS (Aim)

Linear Regression in higher dimensions

$$y = a_1 x_1 + a_2 x_2 + b$$

to fit a linear regression model here:, Need to specify 3 variables a_1, a_2, b

In higher dimensions, (known as Multiple regression) Must specify coefficients for each feature and the variable b

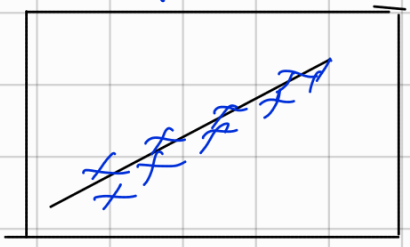
$$y = a_1 x_1 + a_2 x_2 + a_3 x_3 \ldots + a_n x_n + b$$

The default metric of linear regression is R-squared,

R^2: quantifies the variance in target values explained by the features (values range from 0 to 1)

① ⇒ features completely explain the target's variance.

High R^2



Low R^2

Mean Squared error (MSE)	$= 1/n \sum_{i=1}^{n} (y_i - \hat{y})^2$	MSE is measured in target units squared.
Root mean squared error (RMSE)	$= \sqrt{MSE}$	Measure RMSE in the same unit at the target variable.

Cross-validation motivation

→ model performance is dependent on the way we split up the data

→ Not representative of the model's ability to generalize to unseen data.

→ solution: Cross-validation.

Cross-validation basics

Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 1
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 2
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 3
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 4
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 5

Training Data | Test Data

5 values of R-square. ⟶ default score for linear regression.

5-folds ⟶ 5 fold CV

k-folds ⟶ k-fold CV

more folds = more computionally expensive.

95% confidence interval

↳ If we were to take 100 different samples and compute a 95% confidence interval for each sample, then approximately 95 of the 100 confidence interval will contain the true mean value (μ)

Regularized Regression

Regularization in regression to avoid overfitting.

Linear regression minimize a loss function,
It chooses a coefficient 'a' for each feature variable, plus 'b'

Large coefficient can lead to overfitting, Regularization penalize large coefficients.

Ridge regression

$$\text{Loss function} = \text{OLS loss function} + \alpha \sum_{i=1}^{n} a_i^2$$

Ridge penalizes large positives/negative coefficients.

α: parameter we need to choose.

picking 'α' is similar to picking K in KNN

Hyperparameter, variable used to optimize model parameters

'α' controls model complexity.

$\alpha = 0$ = OLS (can lead to overfitting)

very high α: can lead to underfitting.

Lasso regression

* Loss function = OLS function + $\alpha * \sum_{i=1}^{n} |a_i|$

* Lasso regression for feature selection

→ Lasso can select important features of a dataset
→ shrinks the coefficients of less important features to zero.
→ features not shrunk to zero are selected by lasso.

How good is your model?

* Measuring model performance with accuracy
  - → fraction of correctly classified examples.
  - → not always a useful metric.

class imbalance

* classification for predicting fraudulent bank transactions
  - → 99.1% of transactions are legitimate
  - → 0.9% are fraudulent

* could build a classifier that predicts NONE of the transactions are fraudulent
  - → 99.1% accurate.
  - → But terrible at actually predicting fraudulent transaction.
  - → fails at its original purpose

* class imbalance : uneven frequency of classes
* Need a different way to assess performance.

confusion matrix for assessing classification performance.

← predicted label →

actual label ↑ ↓		
	True -ve	False +ve
	False -ve	True +ve.

True +ve → no. of. Fraudulent transaction correctly labelled.

True -ve → no. of. legitimate transactions are correctly labelled.

False -ve → no. of. legitimate transactions incorrectly labelled.

False +ve → no. of. transactions incorrectly labelled as fraudulent.

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

class of interest → +ve class.

$$\text{Precision} = \frac{\text{true +ve}}{\text{true +ve} + \text{False +ve}}$$

(positive predictive value)

High precision = lower false positive rate.

High precision = not many legitimate transactions are predicted to be fraudulent.

$$\text{Recall} = \frac{\text{True +ve}}{\text{true +ve} + \text{False -ve.}}$$ (Sentivity)

High recall = lower False -ve rate

High recall = predicted most fraudulent transaction correctly.

$$\text{f1 Score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$ (harmonic mean of precision & recall)

↓

equal weight to precision and recall.

Logistic Regression and the ROC curve

Logistic regression is used for classification problems.

Logistic regression outputs probabilities

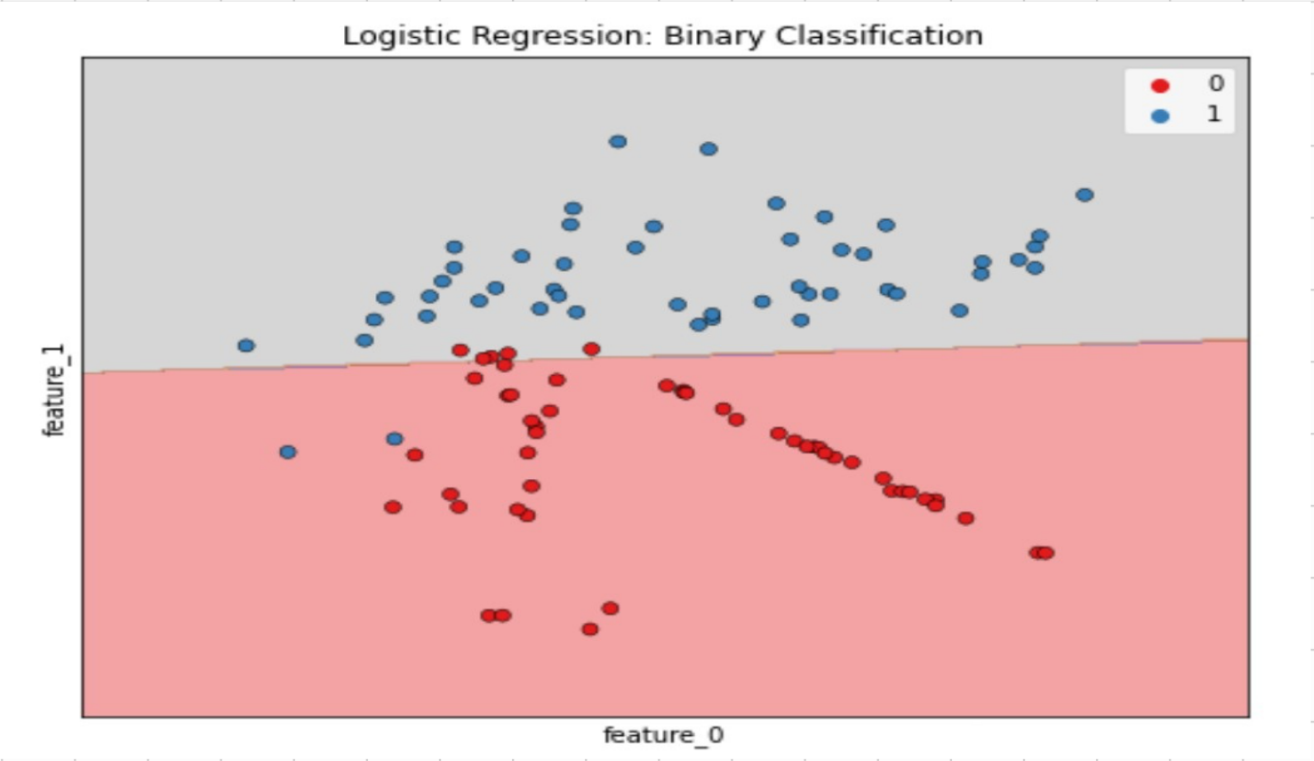
if the probability, $p > 0.5$:

the data is labelled 1

if the probability, $p < 0.5$:

the data is labelled 0

Logistic regression produces a linear decision boundary.



By default, logistic regression threshold = 0.5

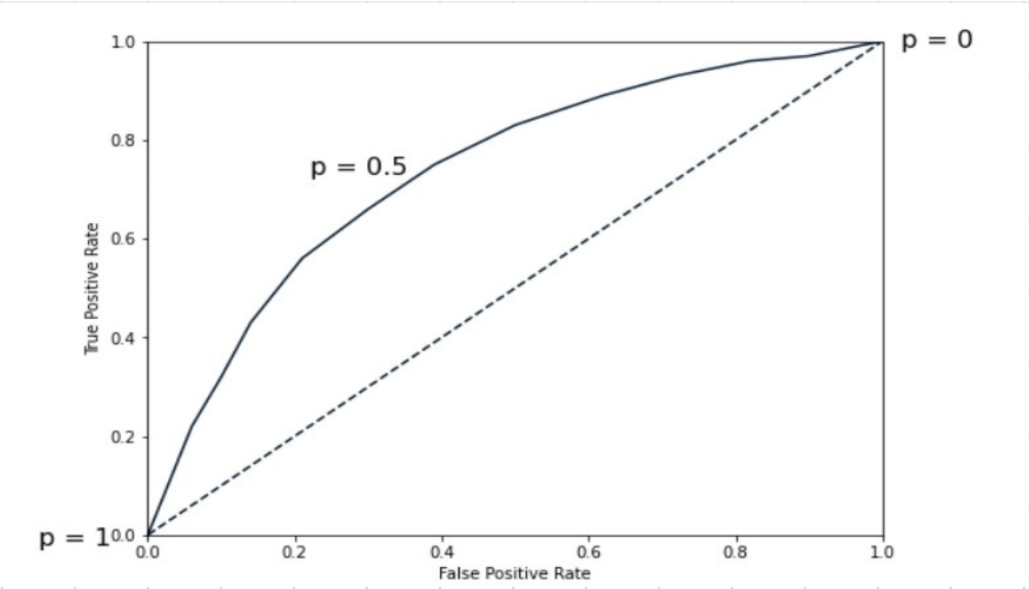
Not specific to logistic regression,
KNN classifier also have threshold.

vary threshold?

Receiver Operating characteristic (ROC) curve, visualize how different threshold affect True +ve and false +ve rates.

ROC curve

If we have a model with '1' for True +ve rate and '0' for false +ve rate, would be perfect model.



AUC
(Area under the curve)
(0 - 1)

→ Measure of the ability of a binary classifier to distinguish between classes is used as a summary of the ROC curve

Higher the AUC, the better the model's performance at distinguishing between +ve & -ve classes.

Hyper parameter tuning

→ Ridge / Lasso regression: choosing alpha.
→ KNN : choosing n_neighbors

→ Hyperparameters: parameter we specify before fitting the model like (α & n-neighbors)

Choosing the correct hyperparameter.

1. Try lots of different hyperparameter values
2. fit all of them separately
3. See how well they perform
4. Choose the best performing value.

↳ hyperparameter tuning.

It is essential to use cross-validation to avoid overfitting to the test data.

We can still split the data & perform cross validation on the training set

We withhold the test data for final evaluation.

Hyperparameter tuning ⟶ Grid search. (Grid of hyperparameter)

Limitations and an alternative approach

- 3-fold cross-validation, 1 hyperparameter, 10 total values = 30 fits
- 10 fold cross-validation, 3 hyperparameters, 30 total values = 900 fits

Alternate way, RandomizedSearchCV, which picks random hyperparameter values rather than exhaustively searching through all options.

ROC is a graph showing the performance of a classification model at all classification thresholds.

Preprocessing Data

Requirement

→ Numeric data

→ No missing data

With real world data

- This is rarely the case
- We will often need to preprocess the data first.

* Scikit-learn will not accept categorical features by default
* need to convert categorical features into numeric values

* convert to binary features called dummy variables.

Dummy variables

genre
Alternative
Anime
Blues
Classical
Country
Electronic
Hip-Hop
Jazz
Rap
Rock

⇨

Alternative	Anime	Blues	Classical	Country	Electronic	Hip-Hop	Jazz	Rap	Rock
1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1

Scikit-learn : OneHotEncoder()

pandas : get_dummies()

Handling Missing Data

Missing data:

* No value for a feature in a particular row

* This can occur because
  - → There may have been no observation.
  - → This data might be corrupt.

* We need to deal with missing data.

A common approach is to remove missing observations accounting for less than 5% of all data.

Imputing values

* Imputation, use - subjective matter expertise to replace missing data with educated guesses

* common to use the mean
* can also use the median or another value.
* for categorical values, we typically use the most frequent value (the Mode)

* Must split our data first, to avoid data leakage (test set information to our model)

Due to their ability to transform our data, imputers are known as transformer.

Imputing within a pipeline

which is an object used to run a series of transformations and build a model in a single workflow.

Data imputation is one of several important preprocessing steps for Machine learning.

Centering & Scaling of data

Scale our data?

→ Many models use some form of distance to inform them

→ features on larger scales can disproportionately influence the model.

→ Ex: KNN uses distance explicitly when making prediction.

→ we want features to be on a similar scale.

→ Normalizing or Standardizing (Scaling & centering)

How to Scale our data:

* Subtract the mean and divide by Variance
  → All features are centered around zero and have a variance of one

  → This is called standardization.

* Can also substract the minimum and divide by the range.
  (Minimum zero & Maximum one)

* Can also normalize so the data ranges from -1 to 1

Evaluating multiple models

→ different models for different problems

Some guiding principles

- Size of the dataset
  - Fewer features = simpler model, faster training time
  - Some models require large amounts of data to perform well
- Interpretability
  - Some models are easier to explain, which can be important for stakeholders
  - Linear regression has high interpretability, as we can understand the coefficients
- Flexibility
  - May improve accuracy, by making fewer assumptions about data
  - KNN is a more flexible model, doesn't assume any linear relationships

It's all in the metric

* Regression model performance
  - → RMSE
  - → R-Squared

* classification model performance.
  - → Accuracy
  - → confusion matrix
  - → precision recall, f1-score.
  - → ROC AUC

* Train several models and evaluate performance out of the box.

Scaling

Models affected by scaling
- → KNN
- → Linear regression (Ridge, Lasso)
- → Logistic regression
- → Artificial Neural Network.

Best to scale our data before evaluating models.